'8080 CPU MODEL 471

The 471 CPU is a complete 8-bit central processor on a single printed circuit board. Based on the 8080A chip, the 471 generates the essential control signals for a complete MIKE 3 microcomputer system.

The 471 supports the full capabilities of the 8080A, including high speed 2us instruction cycle and memory addressing of up to 64K bytes. The instruction set supports logical, arithmetic, and bit-manipulating operations; the CPU can convert data to decimal (BCD) notation with a single instruction. Logical operations set flags which may be tested by conditional branch instructions to control program flow. For added convenience in memory addressing, six general-purpose internal registers may be addressed not only singly, but in pairs. There are also double-precision (16 bit) additions and increment/decrement instructions.

The 8080 stack pointer addresses external RAM memory, allowing for practically unlimited nesting of program subroutines. Through single-byte PUSH and POP instructions, registers, flags, and the program counter can be saved in the external stack, and restored later-- greatly facilitating interrupt handling. The 471 includes circuitry for three interrupt levels, expandable to an eight-level prioritized interrupt system by adding an accessory circuit board. A digital timer restarts the computer automatically should the CPU fail to recognize interrupt requests for a set period.

Power bus drivers are included for driving expanded systems. However, control by the 8080A of the data and address busses may be suspended by a HOLD request, thereby allowing for direct memory access (DMA) or multi-processor operation.

The 471 includes a reliable crystal-controlled clock. Specialized control signals are developed on the CPU board and are present on the system bus to simplify expansion, by minimizing the extra interface circuitry needed on each new memory or I/O board. These signals not only include those required for 8080-type I/O instructions, but also support the 6800/6502 and 8008 I/O addressing modes. The 50-pin bus structure is compatible with these and other eight-bit processors, allowing the 471 to be used interchangeably with other CPU boards in interfacing to memory and accessory modules in the modular micro series.

While supporting all the potential of the powerful 8080A microprocessor, the Model 471 CPU board is a good example of the flexibility of the modular micro approach. Optimal for small systems-- yet fully expandable.

TECHNICAL DATA

SEC. 1 CIRCUIT DESCRIPTION

1.1 INTRODUCTION The 471 CPU board is based on the 8080 microprocessor. Normally, the most complex element in a MIKE 3, the CPU generates nearly all of the control signals which appear on the system's bus structure.

To form a working microcomputer, the 471 board must interfaced with memory storage. For example, The Model 423 PROM/RAM board includes space for PROM memory from which to read permanently-stored program instructions (such as the MIKE 3 Monitor), and RAM memory for temporary data storage and for the push-pop stack. In addition, a complete microcomputer usually includes input/output boards for interfacing with keyboards, displays, printers, and other peripheral devices.

This circuit description conforms to the breakdown of the schematic diagram (page 4 etc) as follows

1.2 8080 CPU
1.3 Bus Drivers
1.4 Main Timing
1.5 System Control Logic
1.6 I/O instruction Modes
1.7 Reset Circuitry
1.8 Interrupt Logic
1.9 Hold Circuitry
1.10 Wait Circuitry
1.11 Indicator Lights
1.12 Power Requirements

1.2 8080 CPU The 8080 is a microprocessor within a single integrated circuit. Based on 17-volt N-channel MOS technology, its 40-pin package includes terminals for an eight-bit bidirectional data bus, D7-D0; a sixteen-bit address bus, A15-A0; power supplies of 12V, 5V, and -5V; and a variety of control signals. The 8080 requires an externally-generated two-phase clock (PH1 and PH2) driven to 12V; all other control signals are at TTL voltage levels, between 0 and 5 volts.

Complete functional and electrical characteristics of the 8080 are provided in the manufacturers' manuals and spec sheets. Essential details are reprented by MR in data sheet DS2204. The reader should refer to these sources for an explanation of internal processor operations, and for detailed information of the 8080 instruction set. This data sheet focuses on the circuitry of the Model 471, a CPU board in the modular micro series from MR.

NOTE CONCERNING 8080A VERSION-- The 8080A is an improved version of the 8080, but is compatible with early 8080 devices. Changes implemented with the introduction of the 8080A include the following: a relaxation of the minimum clock input high voltage from 11.0 to 9.0V; improved current sinking capabilities on output terminals, from 1.7 ma on the data bus (0.75 ma at other outputs) to 1.9 ma on all output terminals, measured at 0.45V; and internal synchronization of the HOLD signal, avoiding the need for synchronizing circuitry outside the 8080.

The 471 CPU board has been designed for use with the Intel C8080A, Advanced Micro Devices Am9080A, or NEC Microcomputers uPD8080A. The Texas Instruments TMS 8080, and early Intel C8080, will work with the 471 CPU board; however, please note the following. The 8224 clock driver circuit is not guaranteed to deliver PH1 and PH2 clock voltages of greater than9.4V, while the TMS 8080 and C 8080 are specified to require 11.0V. Lab tests by MR indicate that i in practice these components do interface successfully. Nevertheless, wecannot interpose a guarentee to that effect where the manufacturer's specifications fail to do so.

The NEC Microcomputers uPD8080A includes two features not found on other versions of the 8080; first, the DAA (decimal adjust accumulator) instruction successfully converts binary data to BCD after a subtraction (as well as addition). Second, a MOV r,r (register-to-register transfer) instruction takes four (rather than five) machine cycles to execute. This last feature improves uPD8080A speed, but means that software is not 100% compatible with other 8080 processors. That is, slight differences in programming will be necessary in programmed timing loops which depend for accuracy on an exact count of machine cycles executed.

1.3 BUS DRIVERS Like other MOS microprocessors, the internal circuitry of the 8080A does not have the power to drive the large array of memory and other devices typically connected to the modular micro bus. For this reason, the 471 includes bus drivers both on the data bus and on the address bus.

The data bus driver stages include two type 8T26 inverting bus drivers. Data traveling on the modular micro data bus is inverted in computers using the 471 Cpu board. Electrically, this improves system noise immunity. Accessory memory boards using PROM memory normally alsouse inverting bus drivers, so inversion of data when programming PROMS is unnecessary. In the case of RAM, there is no problem even when non-inverting bus drivers are used on the RAM board; if data is inverted when written into RAM, it is reinverted (and restored) when read back. Input and output data, however, is affected.

The DBIN signal, produced by the 8080A, indicates that the CPU is ready to read data from memory or an input device. Inverted by the systen control logic (Sec. 1.5) DBIN/NOT activates the inward-pointing drivers of the two 8T26 ICs. The outward-pointing drivers are activated by SYNC and WR/NOT, both CPU-produced signals--the first indicating that machine status information is present on the bus. When neither pin 1 nor pin 15 is activated, the data terminals of the 8T26s are in a high-impedance state, and do not load the data bus.

The Address/Control Bus driver stage includes four type 8T97 hex tri-state bus drivers. All 24 driver circuits are normally active, driving the 16-line modular micro address bus, as well as eight selected control lines. Only when the CPU has acknowledged a HOLD request are these drivers deactivated, allowing some device other than the 471 CPU board to control the system. Useful DMA and multi-processor applications, this provision is discussed further in Sec. 1.9.

SEC. 1.3 BUS DRIVERS (con't)

In the modular micro series, the 16 address lines are designated DH7 through DH0 (corresponding to A15 through A8 on the 8080 package), and ~~DH7~~ DL7 through DL0 (A7 through A0).

The 8T26s can sink up to 40 mA of current (about 25 standard TTL loads), and the 8T97s, up to 48 mA (about 30 standard TTL loads)--more than enough for most microcomputer systems. In addition, both bus driver types are assured of delivering a logic high input voltage of at least 3.5V to the 8080 (at rated 8080 leakage currents), meeting the 8080 V_{IH} specification of 3.3V.

1.4 MAIN TIMING The 8080 requires a two-phase clock referenced to 12V. The standard 8080 requires a clock cycle time t_{cy} ranging between 480ns and 2.0 us;The 471 runs with a 500 ns clock cycle time, for an instruction cycle time of 2.0 us. An instruction cycle is defined for speed spec purposes as the time required for the 8080 to execute an instruction consisting of a single machine cycle of four states, each state lasting one clock period. Actual instruction cycles vary in duration between four and 18 clock periods, depending on the instruction involved. The number of clock periods is further increased if the CPU is made to wait for slow memory.

The 471 CPU board uses a type 8224 clock generator/driver for supplying the 8080 with the required PH1 and PH2 signals. An 18.0 MHz crystal is connected to the 8224, whose internal oscillator produces an 18 MHz square wave (available at pin 12 for test purposes). An internal divide by nine counter produces two alternating positive pulses with a basic frequency of 2;0 MHz and an asymmetrical duty cycle (see Intel 8080 Manual). After passing through high-level drivers, these signals are connected directly to the 8080, as PH1 and PH2. A TTL version of PH2 -PH2T- is also generated, and is available for system control purposes.

At press time, available high-speed versions of the 8080A (with maximum clock periods shown in nanoseconds) included Intel C8080A-2, Advanced Micro Devices Am9080-2 (380); C8080A-1, Am9080-1 (320); Am9080A-4 (250). The Intel 8224

can be used to generate 8080 clock signals up to a maximum frequency of 27 MHz (corresponding to a clock period down to 370 ns), which supports the 8080A-2. The 8224 (AM) is specified to function with CPUs up to the 250 ns clock cycle speed. (These figures were those available at press time; new product releases were scheduled, and the reader should check for current specifications.)

In order to permit the use of an overtone crystal which may be more economical at frequencies above 20 MHz--a tank circuit may be added, as shown by the optional parts connected to the 8224 in the 471 schematic. L1 and C12 are chosen for resonance at the overtone frequency, and C1 is a RF by pass capacitor.

The 8224 also supports three system control functions. First it generates STB/NOT, a status strobe occurring at the beginning of each 8080 machine cycle. STB/NOT is used to strobe system control logic (Sec. 1.5), and is made up of SYN (an 8080 signal indicating the beginning of a machine cycle) and Ø1A, which occurs "1 unit" before Ø 1 proper (referring to Intel timing). Using Ø1A and incorporating this logic in the Schottky bipolar circuitry of the 8224 optimizes system speed. The 8224 circuitry also activates STB/NOT during reset cycles.

The 8224 synchronizes the system reset signal at its RES/NOT input with Ø2D of the clock (the rising edge of Ø2), to provide the synchronization required by the 8080. (For more on the reset function see Sec. 1.7 below).

Finally, the 8224 synchronizes the microcomputer's BSY/NOT line with Ø2D of the clock, again to meet 8080 internal timing requirements. (For more on the use of the BSY/NOT line for memory wait cycles, see Sec. 1.10.)

1.5 SYSTEM CONTROL LOGIC The 8080 is a synchronous device which follows certain intricate sequences in executing program instructions stored in memory. An instruction cycle is required for the execution of each instruction, with this time period subdivided into specialized cycles, as required in order to time-share the microprocessor's eight-bit data bus and 16-bit address bus. An instruction cycle consists of between one and four machine cycles, with the first always devoted to fetching the computer's program instruction from memory. The address bus selects the instruction location, and the data bus receives the memory word. In general, one additional machine cycle is necessary each time the instruction writes or reads memory or an I/O prot .

Near the beginning of each machine cycle, the 8080 produces the signal SYNC, which is gated by the 8224 clock generator to produce the system strobe STB. At STB time,

the 8080 data bus contains an eight-bit status word which defines the purposes of the machine cycle. The assignment of data bits is as follows:

D7	MEMR	Memory to be read from
D6	INP	Input data to be received
D5	M_1	First byte of instruction to be fetched
D4	OUT	Output data to be transmitted
D3	HLTA	CPU in the HALT state
D2	STACK	Stack operation
D1	WO/NOT	Write into memory or output port (active low)
DØ	INTA	CPU acknowledges an interrupt request

The system control logic on the 471 CPU board includes a 74LS174 hex latch and a 74LS74 D flip flop. Both circuits are strobed by STB (STB/NOT inverted), latching up seven of the eight bits of the 8080 status word. (The M_1 bit is not used.) Note that the inputs to the latches ~~/////////~~ derive from the inverted data bus, DB7/NOT through DBØ/NOT; the data terminals of the latches have been bubbled to indicate the active-low logic involved. Three of the resultant outputs are inverted and gated with DBIN to produce MEMRA/NOT, IORA/NOT, and INTX/NOT. DBIN comes from the 8080 (through buffering inverters), and indicates that the 8080 data bus terminals are ready to receive from external circuitry. INTX/NOT is used to read an interrupt jam instruction, during interrupt cycles (Sec. 1.8). MEMRA/NOT and IORA/NOT are buffered by the Address/Control Bus Driver stage (Sec. 1.3) and are relabeled MEMR/NOT and IOR/NOT. Distributed via the modualr micro bus to circuitry outside the 471 CPU board, these signals are used for enabling memory to be read from, and input ports to be read from, respectively.

The 8080 creates an enable signal called WR/NOT, indicating that it has a valid data byte on the data bus to be written either into memory or into an output port. To distinguish between these write operations, two NAND gates combine WR (WR/NOT inverted) with the status word bit OUT, Which is latched up in the 74LS74 and is available both true and complimented at the flip flop outputs. STEWA/NOT (sometimes called MEMWA/NOTØ is active during memory write cycles, and IOWA/NOT during output write cycles. Both are buffered by the Address/Control Bus Drivers and appear on the modular micro bus as STEW/NOT and IOW/NOT. Note that these are enable signals, valid only at the moment when the 8080 is transmitting data on its data bus.

The signal MEMA/NOT is produced by ORing together MEMRA/NOT and STEWA/NOT, and after buffering, is available as MEM/NOT ON the modular micro bus. It is a combined strobe/enable signal, active during cycles referring to memory (read or write), at the moment when data is transferred between CPU and memory via the data bus.

The signal PCWA/NOT indicates that a memory write cycle is in progress. Unlike the signals described above, it is not a strobe/enable signal -- that is, it is not valid just at the moment when data is transferred on the data bus. Rather, PCWA/NOT starts near the beginning of a memory write cycle and does not end until the next machine cycle begins. A NAND gate is used to gate WO (a status bit indicating a memory or output write cycle) with OUT/NOT, in such a way that PCWA/NOT is active only when OUT/NOT is inactive. After buffering, the signal is called PCW/NOT, and is available on the modular micro bus. Typically, PCW/NOT is used to control the R/ W/NOT terminals of RAM memory. PCW/NOT anticipates both STEW/NOT and MEMR/NOT, so that the read/write/NOT mode on the RAM can be set up before the strobe/enable signal activates data transfers with the CPU.

Another signal useful in interfacing with memory is the status strobe, STB/NOT, developed by the 8224 clock generator and discussed above. After buffering by the Address/Control Bus Driver stage, it is piped along the modular micro bus under the label ADV/NOT, for address valid. As the name implies, when ADV/NOT occurs the 8080A address lines have been loaded and are stable, indicating that the address bus may be used reliably in memory and I/O addressing. ADV/NOT occurs early enough in the machine cycle that, should the ready line be pulled down at this time, the CPU will enter a wait state (See Sec. 1.10.)

1.6 I/O INSTRUCTION MODES

1.6.1 STANDARD 8080 I/O MODE The IOR/nOT and IOW/NOT signals discussed above are used for strobing and enabling input and output ports, using the standard 8080 INP and OUT instructions. In either of these instructions the second byte is an I/O port address, which appears on both the DH and DL address lines of the microcomputer. The I/O circuitry must first be addressed by the DH or DL address lines; then the IOR/NOT signal enables an input port (or IOW/NOT strobes an output port) and the I/O data flows between the port and the 8080 accumulator (A register).

1.6.2 MEMORY-MAPPED I/O A second mode of I/O addressing is commonly used -- simply defining selected addresses in memory as I/O ports. Output ports are addressed by memory write instructions -- strobed with the STEW/NOT (MEMW/NOT) strobe. Input ports are addressed by memory read instructions -- enabled with MEMR/NOT signals. This memory-mapped I/O

mode is that used in the 6800/6502 microprocessor series, and is attractive because of its universality. That is, the approach is basically compatible with any microcomputer, and a peripheral device with this kind of I/O port addressing circuitry can most likely be interfaced with few modifications to different microprocessors. This is especially true in the modular micro series, where the bus is compatible with various standard eight-bit processors. Another advantage is that the 8080A includes instructions for transferring data between memory and a number of CPU index registers, rather than just the A register (accumulator) as with I/O instructions proper. Since the Computer's memory read/write signals are used in controlling memory-mapped I/O no additional circuitry is required. Naturally the user must maintain a memory map, to make sure that memory and I/O port addresses do not overlap.

By predecoding some of the I/O address lines, producing a specialized I/O strobe/enable signal, the amount of I/O strobe decoding circuitry necessary on peripheral boards in the microcomputer can be reduced. (See Sec. 1.6.4)

1.6.3 8008-COMPATIBLE I/O A third mode of I/O addressing is supported by the 471 CPU board, which derives the DINA/NOT strobe/enable signal (as shown in the System Control Logic section of the schematic). This I/O addressing mode is used by the 8008 microprocessor, the 8080's predecessor, widely used in industrial controllers. An 8008 microcomputer-- the MIKE2-- is available from Martin R, and some boards in the modular micro series use the 8008 mode of I/O addressing. The fact that the 471 includes these features makes it very attractive in updating 8008 systems, especially when it is not desired to change peripheral I/O boards.

In The 8008 I/O mode, the I/O port address appears on five lines of the high-order memory address bus, namely DH5 through DH1. Output data appears on the low-order memory address, DL7 through DLØ. Input data is transmitted to the CPU via the data bus. When executed on an 8008, there are special single-byte input and output instructions; but the Model 471 8080 CPU board, these 32 I/O ports are addressed with certain memory referencing instructions.

Referring to the circuitry which creates DINA/NOT: this signal occurrs when the 8080 data bus is ready to receive data (DBIN/NOT) or transmit data (WR/NOT); when the instruction is not a standard 8080 I/O instruction (OUT/NOT and IN/NOT must be high); and only when DH7 and DH6 are high, and DHØ is high. These conditions mean that any 8080 memory read/write instruction which references the odd-numbered pages in the upper one-fourth of memory can be used as an 8008- comaptible I/O instruction. (Table 1.6.1- not copied).

DIN2A/NOT is a variant of DINA/NOT, differing only in that it is created when DHØ is low (rather than high). By using DIN2/NOT, another set of 32 I/O ports can be addressed.

The Address/Control Bus Driver stage converts DINA/NOT to DIN/NOT, and DIN2A/NOT to DIN2/NOT, for use on the modular micro bus. Note that the user must choose between the extra 32 8008-compatible I/O ports (using the DIN2/NOT signal), and the standard 8080 input ports (using the IOR/NOT signal), by wiring up either JUMPER (J9) or (J10). As the solid line indicates, factory-assembled boards are wired for IOR/NOT for the 8080.....

To decode the strobe and enable signals for an 8008-compatible I/O port, circuitry on a peripheral board combines the DIN/NOT or DIN2/NOT signal on the sodular micro bus with the five address bits DH5 through DH1. Using a single 74LS138 (3205/8205), eight such strobe/enable signals can be produced by connecting the five DH bits in various combinations to the three select inputs, and to one active-low and one active-high enable input. (This technique is elaborated in the data sheet for a Universal I/O Kit, KT9001, and in Chapter 7.

Several kinds of 8080 instructions may be performed using the 471 CPu board in order to address 8008-compatible I/O ports. (Figure 1.6.2, not copied)

Note that many of these 8080 instructions can be used as combined I/O instructions. Assume that a strobe decoder on a peripheral I/O board combines DIN/NOT with some combination of DH5 through DH1-- say 00100, corresponding to octal memory page 311. The resulting signal is used to strobe an eight-bit latch, whose inputs are connected to the DL address bus. This is an output port-- equivalent to 8008 Output port 4. The same signal is used to enable an eight-bit bus driver, with inputs deriving from some eight-bit parallel data source, and tri-state driver outputs connected to the microcomputer data bus (8008 Input Port 4). The 8080 instruction LDA 311 DDD is executed (where DDD is the output data). DIN/NOT GOES LOW: THE H address 311 is decoded; both input and output ports are activated; the output data DDD ends up in the output port; and the input data is moved to the A register.

For example, the program....

```
LXI     B,311123
LDAX    B
```

....will output the octal value 123 to Output Port 4, and simultaneously lead the A register with the data at Input Port 4. The instruction LDA 311123 would accomplish the same thing.

Note also that if the purpose of the instruction is to input data only, the low-order address need not set up. For example, the instruction, LDAX B might be executed, with the input port address previously loaded into the B register, but any arbitrary data in the C.

Finally, note that the use of these 8008-compatible I/O instructions is entirely optional. Any of the page addresses shown may instead be used for memory, activated by the normal 8080 memory signals (STEW/NOT, MEMR/NOT, etc), and DIN/NOT may be ignored. As with other forms of memory-mapped I/O -- of which this is a specialized version-- the only constraint is th t a given memory page address not be used for more than one purpose (memory or I/O port).

1.6.4 MEMORY-MAPPEDI/O WITH PREDECODED STROBE One advantage of the 8008-compatible I/O mode is that there are only five i/o address lines, as opposed to eight with the 8080 mode and eight with the memory-mapped mode (if a whole page in memory is used for each I/O port; 16 lines if a single location in memory is assigned as an I/O port). This means that circuitry needed to produce I/O strobes, for activating input bus drivers and output latches on a peripheral circuit board, must combine eight or more I/O address slines, plus a control signal (IOR/NOT, IOW/NOT, MEMR/NOT, STEW/NOT). This circuitry will be cumbersome, especially if more than one I/O strobe/enable signal is needed on that board.

The user of the 471 CPU board can combine the advantages of several I/Oaddress modes by using DIN/NOT (or DIN2/NOT) for memory-mapped I/O. To output data, the program executes a memory write instruction, and data is written into the output port; to input data, the program executes a memory read instruction, and data is read from the intput port into the CPU. All of the various memory-referencing instructions of the 8080 may be used. So far, this is a description of memory-mapped I/O in general. The only differences are these: The I/O ports use DIN/NOT (or DIN2/NOT) instead of MEMR/NOT or STEW/NOT; and the high-order memory address which references these I/O ports must be one which will cause DIN/NOT (or DIN2/NOT) to be produced.

With this I/O method, a single 74LS138 on a peripheral board will produce eight I/O strobe/enable signals as described above (1.6.3).

1.7 RESET CIRCUITRY When power is first applied to the 8080, its internal registers may take on any random values. To initialize the machine, the 8080 RESET terminal must be brought to a logic high level for at least three clock cycles. This resets the program counter to zero, and the 8080 will proceed to execute stored program instructions, starting at location zero. The reset signal also resets the 8080 interrupt flip flop, disabling interrupts (as does an interrupt itself). An initial interrupt is unnecessary for starting the computer. However, if it is desired to make the CPU wait for an initial interrupt before proceeding with its mainline program, the two instructions EI (enable interrupt), HLT (halt) may be inserted starting at location zero. 8080 flags and registers other than the program counter and interrupt flip flop are not affected by the reset pulse, but must be initialized as required through program.

The Model 471 CPU board includes a power On Reset circuit which senses the rising of the 5V power supply as power is applied, generating the signal RES/NOT. The LM3302/339 is capable of operation at supply voltages as low as 2V. Until the 5V line reaches 3.9V, the zener diode does not conduct, and the -input to the comparator receives the full power supply voltage. The resistive voltage divider, R13/R11, provides the plus comparator input with 9/10 of the power supply voltage; and since the -input exceeds the plus input level, the comparator output goes low, activatin g RES/NOT. When the 5V line reaches about 4.3 volts, the voltage at the plus comparator input finally begins to exceed the 3.9V zener voltage, and the comparator goes high, remaining so as the 5V supply finishes charging up to a full 5V.

The RES/NOT signal is connected to the 8224 clock generator, which synchronizes the reset request with 8080 timing (Sec. 1.4 above). The output of the 8224, RESA, is connected directly to the 8080A CPU.

Should the 5V line ever dip below about 4.3V, the comparator will again go low, after a delay while C8 discharges -- generating RES/NOT and resetting the computer. The delay will prevent short glitches on the power supply from causing a reset. But if the voltage drops much below 4.0V, the diode (DA806) starts to conduct, discharging the capacitor more rapidly, and causing a reset.

Note that this reset circuit looks only at the 5V power supply. For reliable operation of the computer, the 12V and -12 (or -9)V supplies, also powering the 8080, should come up if not before, at least very soon after the 5V power supply does.

An optional reset switch may be installed, allowing the user to reset the microcomputer manually. This provision is generally unnecessary in production versions of the 471 CPU board, but is useful in prototyping. Consider the following potential 8080 system failure; a DI (disable interrupt) instruction has been executed, in order to protect a subrouting which must not be interrupted (such as a critical timing loop or stack operation). Because of a software or hardware bug, a HLT (halt) instruction is executed. This condition is inescapable through interrupts, since the CPU has masked them out. Only a system reset will restart the computer.

As an additional protection against this failure mode, the 471 board incorporates a digital timer which automatically resets the 8080 if the CPU has masked interrupts (through a DI instruction) and fails to acknowledge interrupt requests for a set period. The timer consists of a 74393 dual divide-by-16 counter. The 8080 has an output terminal labeled INTE (interrupt enable) which is high when

interrupts are enabled. When the CPU is in the state where interrupts will be acknowledged, the inverted version of INTE causes a low input to a 74LS00 NAND gate, whose output is therefore high. This holds the 74393 reset (R) terminals at a high level, forcing the counter output bits to zero and inhibiting counting.

When interrupts are disabled, INTE goes low. If at the same time, an interrupt request has been received, but has not yet been acknowledged by the CPU, the 74LS74 interrupt request flip flop output (B5, pin 5) is high. These conditions are combined by the NAND gate, and the 74393 reset signal is removed, allowing the counters to count. Clocked by PH2T, the 74393 counts once each 8080 clock cycle. If the CPU acknowledges a HOLD request (Sec. 1.9), HLDA/NOT goes low, inhibiting counting -- but not resetting the 74393. If the CPU acknowledges an interrupt request, then the counters are reset. But if 128 PH2T cycles have been clocked, the high-order 74393 output bit (pin 8) goes high. The signal which results (after inversion), REE/NOT, causes a reset.

REE/NOT is ORed with the signal RESET/NOT, which originates from an optional control board (Sec. 1.8). A 7417 open-collector buffer is connected to the RES/NOT line, which may also be pulled low by the power-on reset comparator and the optional reset switch.

The automatic reset circuit may be defeated by removing the 74393 IC from its socket, and connecting a wire jumper between pins 8 and 7 (ground) on the 74393 socket.

1.8 INTERRUPT LOGIC The 8080 fully supports external interrupt requests. A typical interrupt source is a peripheral input device (such as a keyboard) which inputs data relatively infrequently and unpredictably, but which must receive immediate priority when ready. Basically, the 8080 receives the interrupt request and issues an interrupt acknowledgment. External circuitry detects the acknowledgment, and jams in a special instruction, causing the CPU to jump to the appropiate interrupt subroutine. When this ends, the CPU returns to the mainline program at exactly the point where it was interrupted.

In many systems there are multiple interrupt sources. It is necessary therefore to provide means for decoding the source of an interrupt request, in order to make the CPU execute the appropiate interrupt instructions. The 471 includes on-board provisions for three interrupt levels. In addition, the essential interrupt control signals are routed to a 20-pin connector at the side of the board, shown as P2 on the 471 schematic. An optional interrupt control board may be connected here for expansion to an eight-level prioritized interrupt system.

The interrupt cycle begins when some device within the modular micro system pulls the INTR/NOT line (pin 34) low. INTR/NOT is connected through jumper J7 to a 74LS20 inverter stage. (J8 would replace J7 only if the optional interrupt board were installed.) The 74LS74 interrupt request flip flop is clocked, causing the INT signal to go high. The INT signal is connected directly to the 8080, requesting an interrupt.

The 8080 contains an internal interrupt flip flop, INTE (as mentioned above). Executing a DI (disable interrupt) instruction, resetting the computer, or performing an interrupt, will reset the INTE flip flop and prevent the CPU from acknowledging further interrupts. To enable interrupts, an EI instruction must be executed. This sets the INTE flip flop and permits the 8080 to recognize interrupts. The 8080 CPU does not itself save interrupt requests which were placed when interrupts were disabled. However, the 471 CPU board does, in the interrupt request flip flop (a 74LS74).

When an interrupt is recognized, the 8080 enters a special interrupt instruction cycle. In the machine status word for the instruction fetch cycle appearing on the data bus at STB time, DB0/NOT will be 0. The system control logic detects this interrupt acknowledgment, and at DBIN/NOT time, produces the signal INTX/NOT (Sec. 1.5 above). This enable signal is used to jam the special interrupt instruction onto the data bus. It also resets the 74LS74 interrupt request flip flop.

INTX/NOT is connected to the inputs of two 7417 open-collector buffers, and to the enable terminals of an 8T97 hex tri-state buffer. The 7417s pull DB7/NOT and DB6/NOT to 0, and the 8T97 pulls DB3/NOT through DB0/NOT to 0. (DB3/NOT is pulled low because pin 12 of the 8T97 is normally low -- unless the optional interrupt board pulls down the NMI line.) DB5/NOT and DB4/NOT are held high by pullup resistors R4 and R3 (which would be replaced by jumpers J2 and J1 only if the optional interrupt board were installed).

These conditions mean that the inverted data bus bits DB7/NOT through DB0/NOT read 00110000. Reinverting, the data word seen by the CPU is 11001111, or 317 octal. This is the code for an 8080 RST 1 instruction, a single-byte call instruction which causes the CPU to jump to memory location 000010, pushing the program counter stack. Starting at this location, the user installs a program which contains the instructions necessary to carry out the purposes of the interrupt request. The interrupt routine ends with a RET (return) instruction, popping the PC stack and returning control to the mainline program.

The The reader should note that, to avoid having the interrupt routine disturb the mainline program, any registers and flags used both by the interrupt and mainline p programs should be saved as the interrupt routine begins (with PUSH instructions) and restored as it ends (with POP instructions).

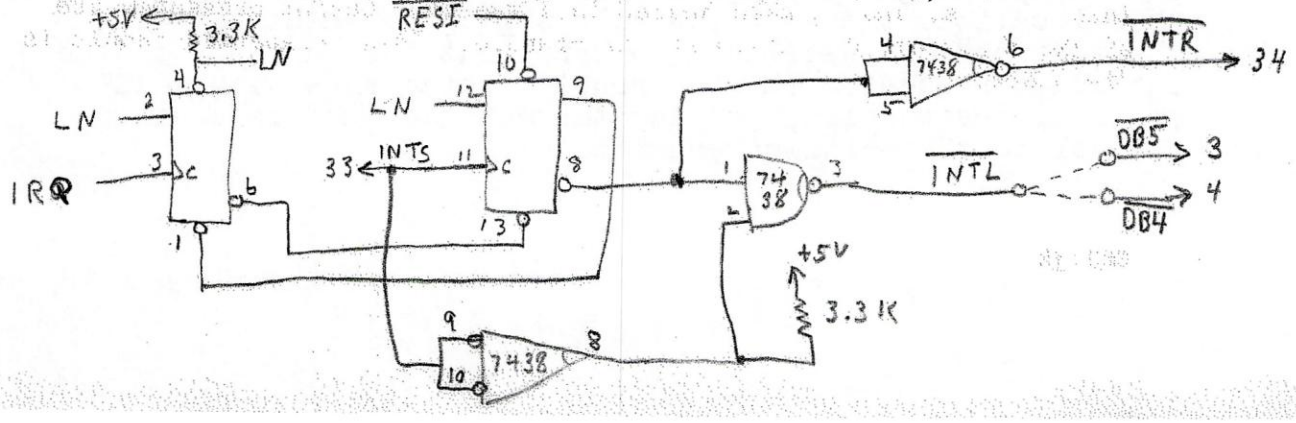
The device requesting an interrupt may pull down the DB5/NOT and DB4/NOT data bus lines at DBIN/NOT time, altering the interrupt jam instruction as shown in Fig. 1.8.1.

$\overline{DB5}$	$\overline{DB4}$	$\overline{DATA}$	DATA OCTAL	INSTRUCTION	RESTART ADDRESS
1	1	00110000	11001111 317	RST 1	ØØ0 010
1	0	00100000	11011111 337	RST 3	000 030
0	1	00010000	11101111 357	RST 5	000 050
0	0	00000000	11111111 377	RST 7	000 070

Fig. 1.8.1 INTERRUPT JAM INSTRUCTIONS

This interrupt system effectively provides three interrupt levels, though there are four possible jam instructions. RST 1 is a general-purpose interrupt, where the interrupting device does not affect DB5/NOT or DB4/NOT. Typically this is a low-priority interrupt. A second interrupt port pulls down DB5/NOT, and a third, DB4/NOT. Other data bus lines could be pulled low but will have no effect unless the system is provided with the optional eight-level interrupt board. Since a RST 7 interrupt jam instruction would result when both the DB5/NOT and DB4/NOT interrupt ports requested interrupts at the same time, it cannot be used unambiguously for a third interrupt port. Note also That if a RST 3, 5, or 7 occurs, it masks the possibility that a RST 1 interrupt might also have been requested.

In order to synchronize circuitry on peripheral boards with the CPU interrupt logic, the INTX/NOT interrupt jam enable signal is connected through jumper J6 to INTS/NOT on the modular micro bus. The device requesting an interrupt uses this signal to enable circuitry which pulls down the DB5/NOT and DB4/NOT data bus lines. A suitable circuit using only two ICs appears in Fig. 1.8.2, below.



The device requesting an interrupt causes IRQ (interrupt request) to go high, setting the first 74LS74 flip flop (pin 3). The Q/NOT output of this flip flop (pin 6) resets the second flip flop (pin 13), causing its output (pin 9) to go low. This signal is fed back to the first FF reset terminal (pin 1), resetting it. However, the second FF remains reset. With pin 8 high, the 7438 open-collector buffer connected to the INTR/NOT line is activated, pulling INTR/NOT low and requesting an interrupt from the CPU board.

The 471 acknowledges with an INTS/NOT enable signal. When INTS/NOT goes low, it activates a 7438 inverter stage ((pins 9, 10, 8). This signal, combined with the high state remaining on the second FF Q/NOT output (pin 8), activates a 7438 open-collector NAND gate (pins 1, 2, and 3), pull-ingdown DB5/NOT, DB4/NOT, or neither, as required. When INTS/NOT ends, going high, the second FF is colcked (pin LL) and set. Pin 8 goes low, ending the INTR/NOT interrupt request, and ending the interrupt jam instruction on the DB5/NOT or DB4/NOT data bus line.

The signal RESI/NOT, which sets the second FF (pin 10), will (when low) disable the interrupt request port. If this signal is not used, it should be connected to logic one (LN).

Naturally the device requesting an interrupt would also have other circuitry for communicating with the CPU. If an input device, the restart instruction would cause the CPU to start executing input instructions addressing the appropriate input ports.

1.9 HOLD CIRCUITRY The 8080 includes provisions for suspending processor operations and allowing some other device to control the microcomputer's data and address busses. Shortly after the 8080 HOLD terminal is brought to a logic high level, the 8080 responds by deactivating its tri-state data and address bus drivers. The 8080 output terminal, HLDA, goes high, acknowledging the hold condition; the inverse of this signal, HLDA/NOT, disables the Address/ Control Bus Driver stages. Since the CPU ceases to create SYNC/NOT, DBIN/NOT, or WR/NOT pulses, the data bus drivers are inactive as well. The 8080 leaves the hold mode when its HOLD input is returned to the logic low condition.

On the 471 CPU board, jumper, J5, connects HOLD/NOT to pin 37 on the modular micro bus. Jumper J4 connects the hold acknowledge signal, HLDA/NOT, to pin 38. (These jumpers would be disconnected if the optional priority interrupt boardwere connected at P2; this board also provides extra DMA channels.)

T To initiate a hold condition, a peripheral board pulls HLDR/NOT low (pin 37). The CPU responds by pulling HACK/NOT low (pin 38). Special circuitry on the peripheral board is then in control of the address bus, the data bus, and the eight control signals shown in the Address/ Control Bus Driver section.

The A version of the 8080 CPU includes internal circuitry for synchronizing a hold request with CPU machine cycles.. If an 8080 (original version) is used, the device requesting a hold must do the synchronizing. This is easily done on the peripheral board by gating the hold request circuitry with ADV/NOT (address valid), pin 26 on the modular micro bus.

1.10 WAIT CIRCUITRY Inspection of the timing diagrams for the 8080 reveals that, operating with a 500 ns clock period, the CPU requires memory with an access time of about 650 ns. However, this is the ideal case, without allowing for delays in bus drivers, address decoders, and other adjunct circuitry in the microcomputer. Intel recommends that the access time of memory interfaced to the 8080 be in the range of 450 to 550 ns, or less. 550 ns is probably a safe figure, but the conservative designer may choose to specify memory with a 450 ns access time. 450 ns is MR policy with regard to the Model 405 4K RAM board.

If slower memory is used, the 8080 must be made to wait for memory access. This is done by pulling the 8080 READY terminal low. When slow memory is addressed, special memory wait circuitry is triggered, pulling down the BSY/NO T (busy) line on the modular micro bus. To synchronize wait requests with the 8080,BSY/NOT is connected to the 8224 clock generator, which drives the 8080 READY terminal (Sec. 1.4).

Any perpheral device with slow memory can pull BSY/NOT down and cause an extended wait period. The same provision may be used with slow I/O devices, during input and output instructions. Optional wait circuitry is a feature of several MR circuit boards.

The 471 CPU board includes a wait circuit consisting of a 9601 or 74122 one-shot and a 7417 open-collector buffer. The one-shot is triggered by STB -- a status strobe which comes sufficiently in advance of any read/write data transfer to quarantee that the CPU will enter the wait state. The Q/NOT output (pin 6) goes low, activating the 7417 and pulling BSY/NOT low.

The 9601 and 74122 are not identical, but both work in this circuit. They have very similar timing characteristics. The values of R1 and C2 determine the length of the wait cycle. In factory-assembled boards, R1 is 10K, and C2 is 270 pf, producing the required one-shot period of about 915 ns. The goal is to cause two memory wait cycles which adds two clock cycles (2x 500ns) to the normal T3 state, adding to a total of 1.5 us. This compensates for

memory with an access time as slow as 1.55 us.

For this CPU circuitry (8080 with 8224), to cause n wait states, the BSY/NOT line should be pulled low (starting at the leading edge of STB0 for the period

$$(n - 1/6)\ T_{cy},$$

where T_{cy} is the clock cycle time. This formula allows enough latitude to permit using 10% tolerance R and C values.

Greater system speed will result if memory wait cycles occur only when necessary, when slow memory devices are being addressed. It is more efficient to install the wait circuitry on any memory (or I/O) board that requires it, and to use the minimum effective delay in every case. Then the memory wait circuit on the 471 CPU board is deactivated, simply by removing the one-shot from its socket.

1.11 INDICATOR LIGHTS The 471 CPU board has two light-emitting diodes which indicate machine states. A red LED glows when memory wait cycles are being performed; the LED is driven by the 8080 WAIT output terminal, through an inverter and open-collector buffer.

Another red LED glows when the 8080 is in the halt condition. The signal HLT/NOT is derived by the system control logic (Sec. 1.5), driving the LED through a 7417. In order to end the halt, an interrupt must be requested (Sec. 1.8) or the machine must be reset. Recall that if the 8080 interrupt FF is reset, it will not recognize interrupts, and only a reset will end the halt condition.

1.12 POWER REQUIREMENTS The Model 471 board requires 12V at 75 mA maximum, 5V at 600 mA max, and -12V at 7 mA. By removing the 1K resistor at R5 (CPU section of the schematic), and installing it as R6, the board may be operated from a -9V supply rather than -12; the current drawn will be 4 mA.

Pins 39 through 50 on the modular micro bus are assigned to power supply connections.

A

1 — 9601/74122

.25 2 — 7417

3 — LM3302/339

B

1 — 8T97 .67

2 — 8T26 .69

3 — 8T26

4 — 74LS174 1.06

5 — 74LS74 ~~1.06~~ .39

C

1

2 — 8T97 .67

3 — 8080

4 —

5 — 74LS00 .28

D

1

2 — 8T97 .67

3

4

5 — 74LS04 .29

E

1

2

3 — 8T97 .67

4 — 8224

5 — 74393

F

1

.67 2 — 8T97

.28 3 — 74LS00

4

.28 5 — 74LS00

RESISTORS

1 — TIMING
2 — 3.3K
3 — 3.3K
4 — 3.3K
5 — 1K
6 — 1K
7 — 3.3K
8 — 3.3K
9 — 3.3K
10 — 3.3K
11 — 1 MEG
12 — 1K
13 — 100K
14 — 100 Ω
15 — 1K
16 — 100 Ω
17 — 100 Ω
18 — 3.3K
19 — 3.3K
20 — 3.3K
21 — 3.3K
22 —

CAPS

1 — .22 mf
2 — TIMING
3 — .05
4 — .05
5 — .05
6 — .05
7 — .05
8 — 0.1
9 — .05
10 — 10 pf
11 — .05
12 — OSC TANK
13 — .05
14 — .05
15 — OSC TANK
16 — .05

H

1 —

2 — 74LS08 .29

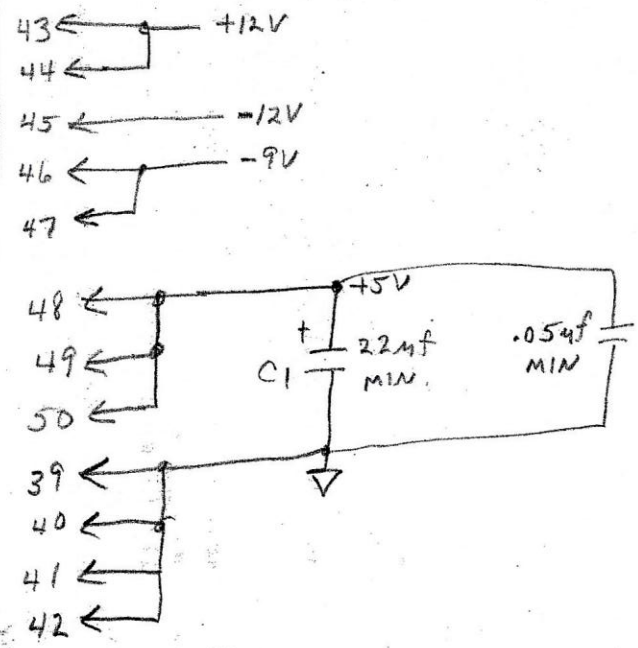
3 — 74LS04 .29

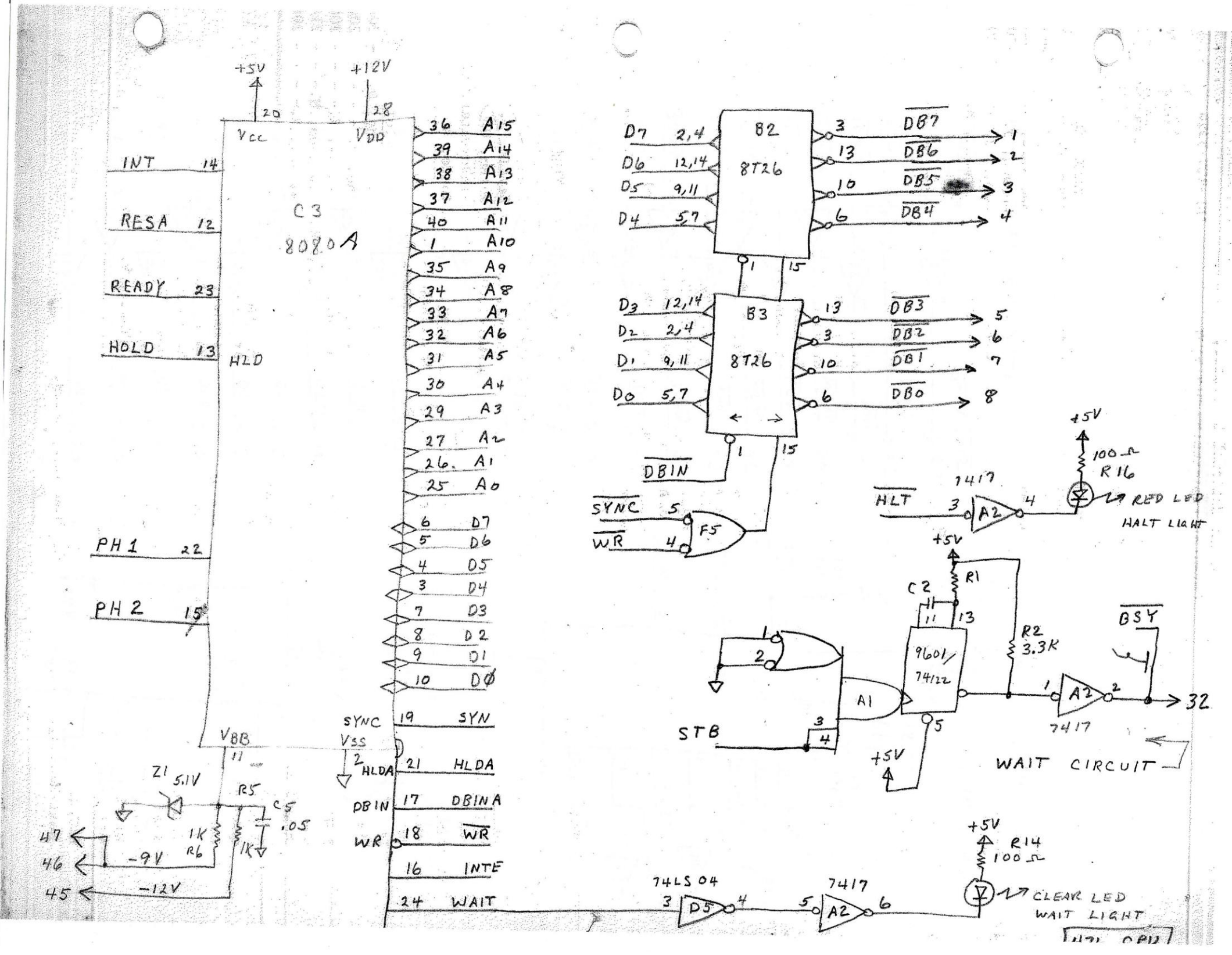
4 — 74LS20 .28

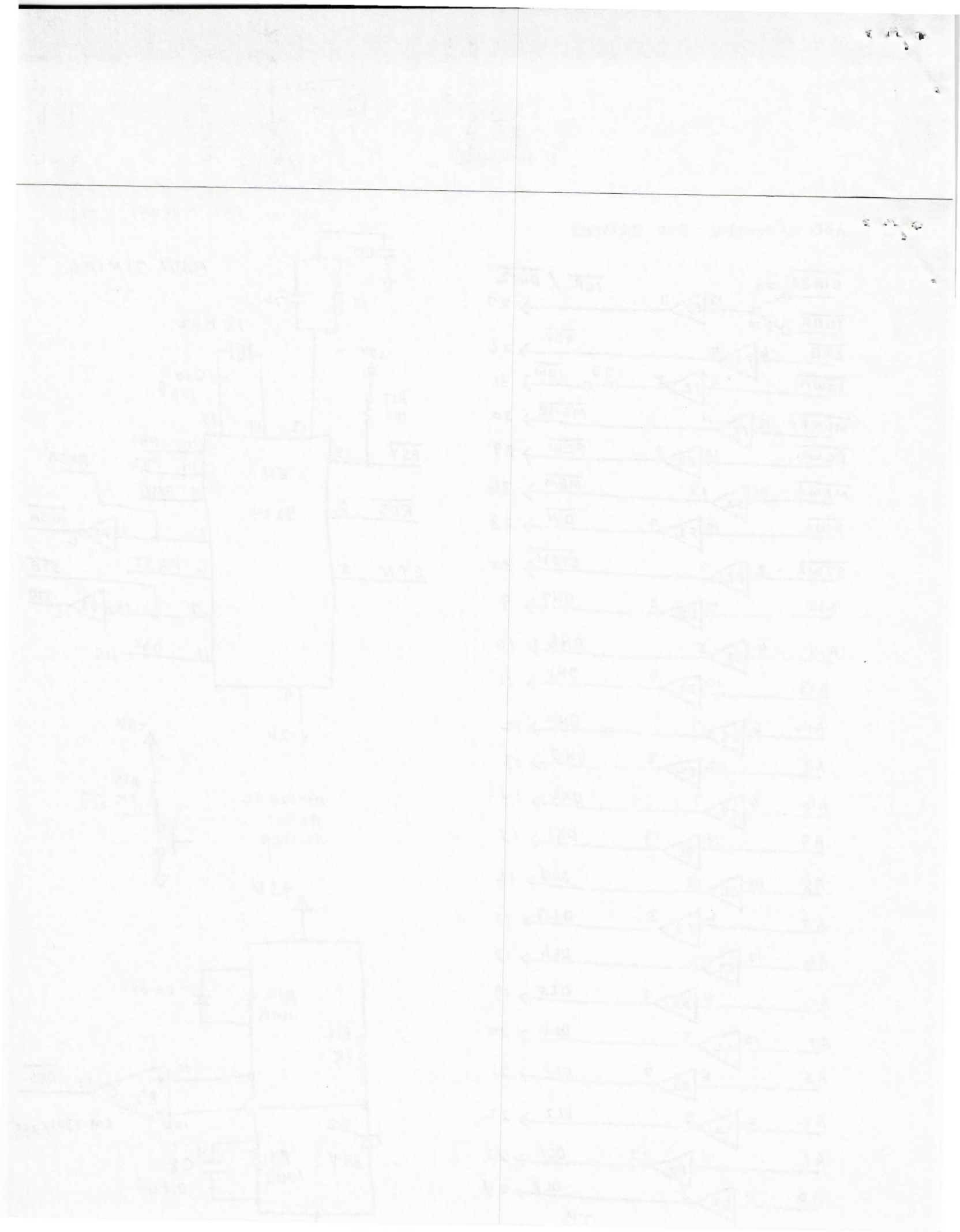
5 — 74LS04 .29

471 POWER BUS

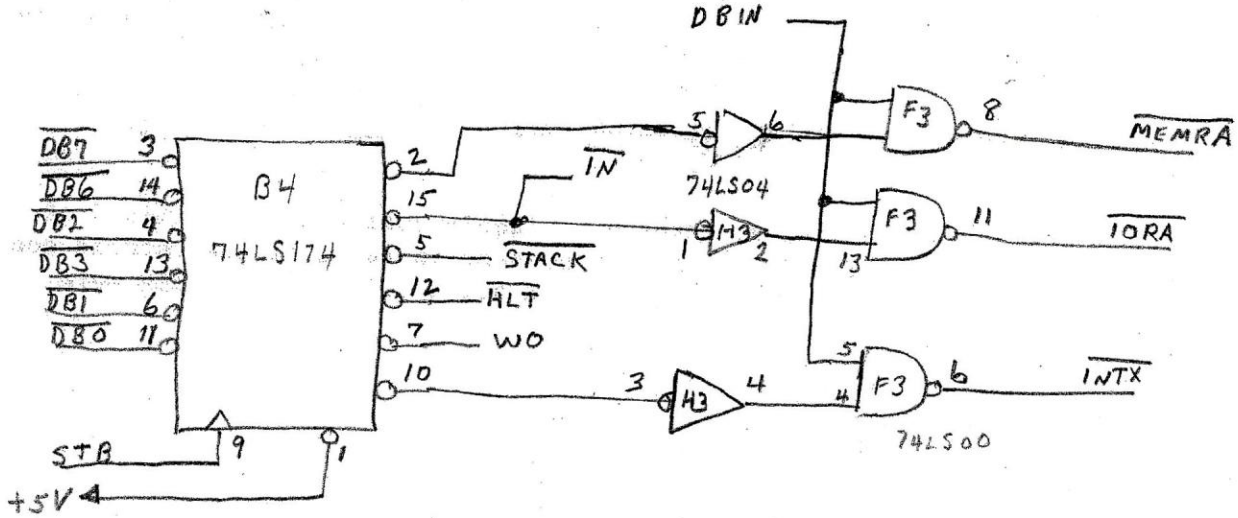
CHIP	+5	GND
3302/339	3	12
	16	8 } others
	14	7 }

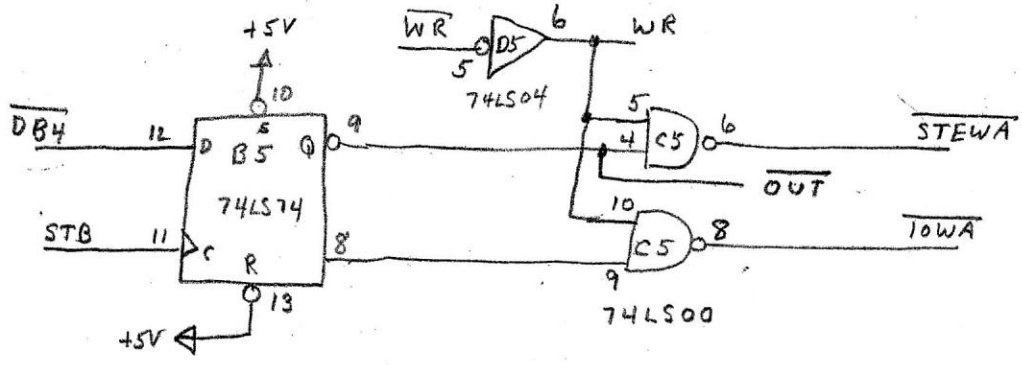


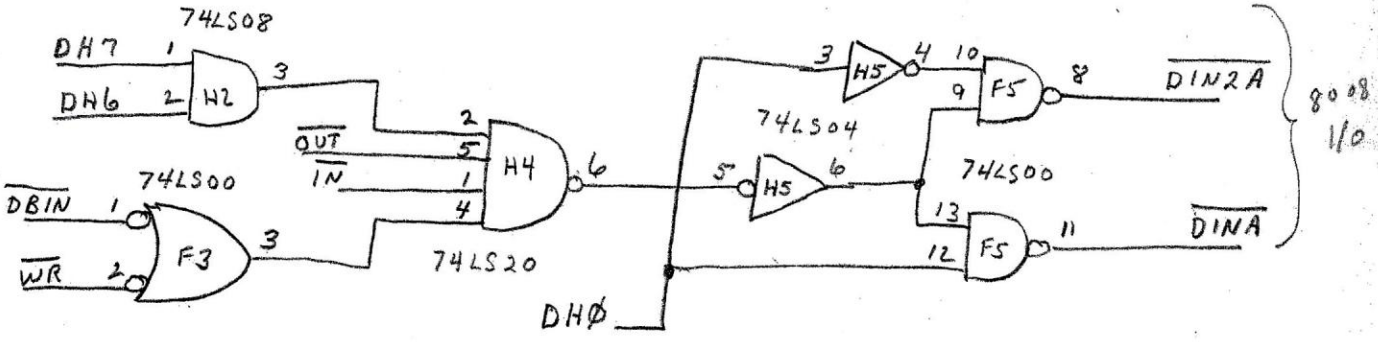

+5V
+12V
20
28
Vcc
VDD
C3
8080A
INT 14
RESA 12
READY 23
HOLD 13
HLD
PH 1 22
PH 2 15
36 A15
39 A14
38 A13
37 A12
40 A11
1 A10
35 A9
34 A8
33 A7
32 A6
31 A5
30 A4
29 A3
27 A2
26 A1
25 A0
6 D7
5 D6
4 D5
3 D4
7 D3
8 D2
9 D1
10 DØ
SYNC 19 SYN
VBB 11
VSS 2
HLDA 21 HLDA
DBIN 17 DBINA
WR 18 WR
16 INTE
24 WAIT
Z1 5.1V
R5
C5 .05
1K R6
1K
-9V
-12V
47
46
45
D7 2,4
D6 12,14
D5 9,11
D4 5,7
B2
8T26
3 DB7
13 DB6
10 DB5
6 DB4
1
2
3
4
15
D3 12,14
D2 2,4
D1 9,11
D0 5,7
B3
8T26
13 DB3
3 DB2
10 DB1
6 DB0
5
6
7
8
DBIN
SYNC 5
WR 4
F5
+5V
100Ω R16
7417
HLT 3
A2 4
RED LED
HALT LIGHT
+5V
R1
C2
11
13
9601/
74122
R2
3.3K
BSY
1
2
A1
3
4
STB
1
A2
2
7417
32
+5V
5
WAIT CIRCUIT
+5V
R14
100Ω
74LS04
3
D5
4
5
7417
A2
6
CLEAR LED
WAIT LIGHT

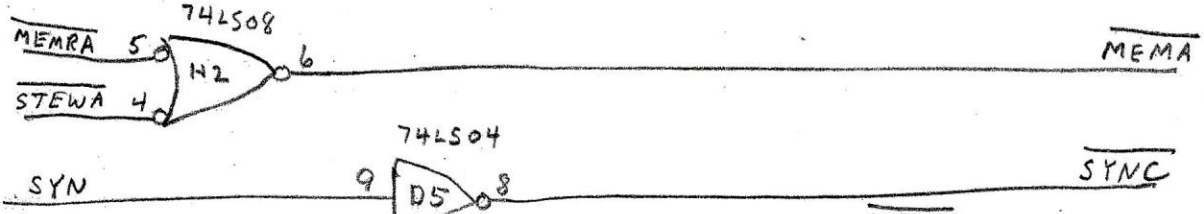


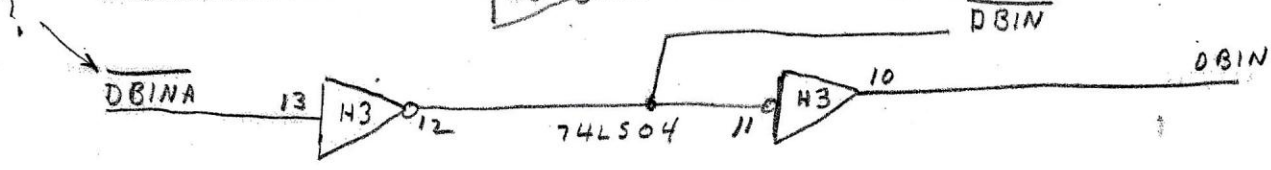
SYSTEM CONTROL LOGIC



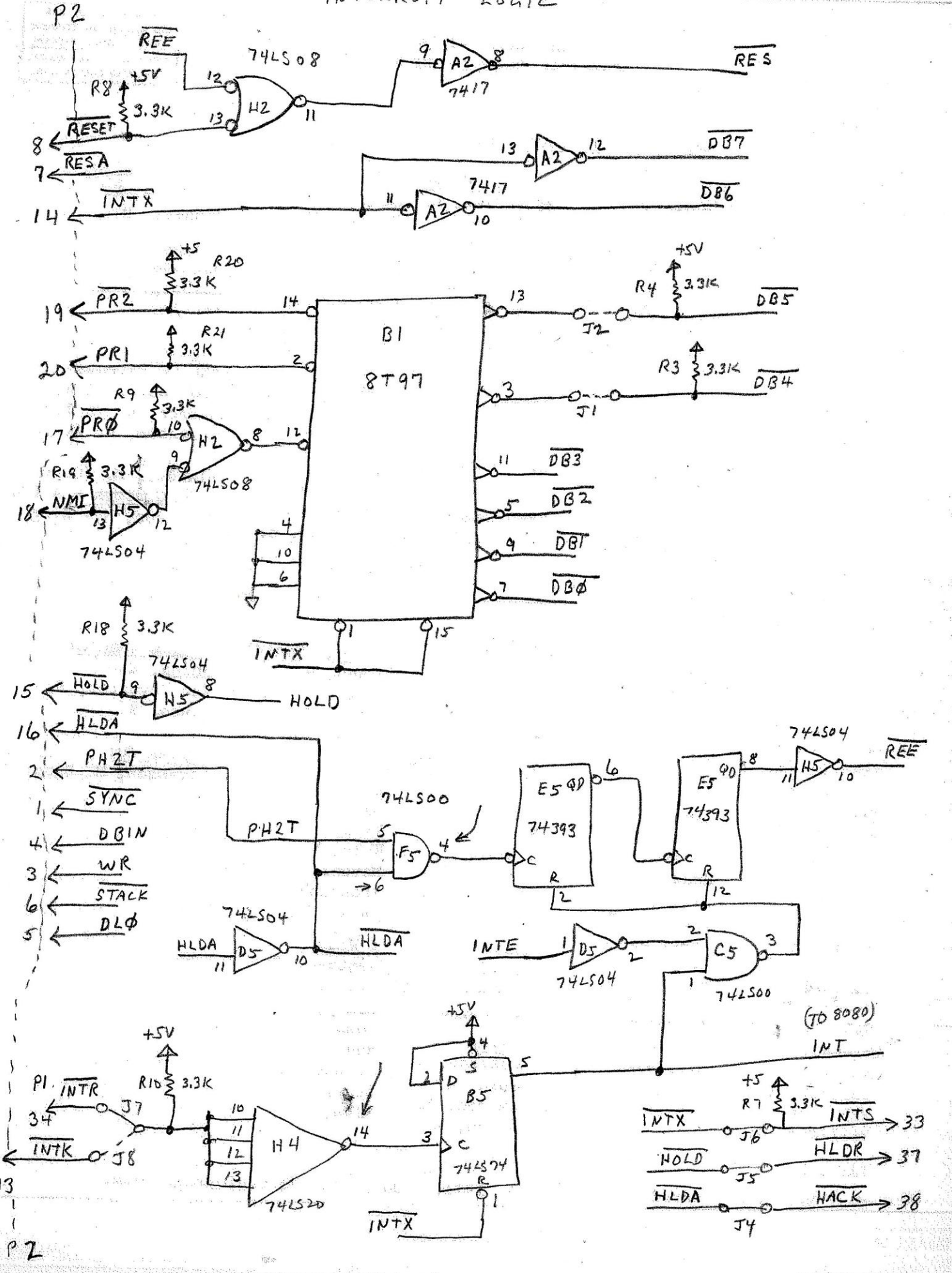




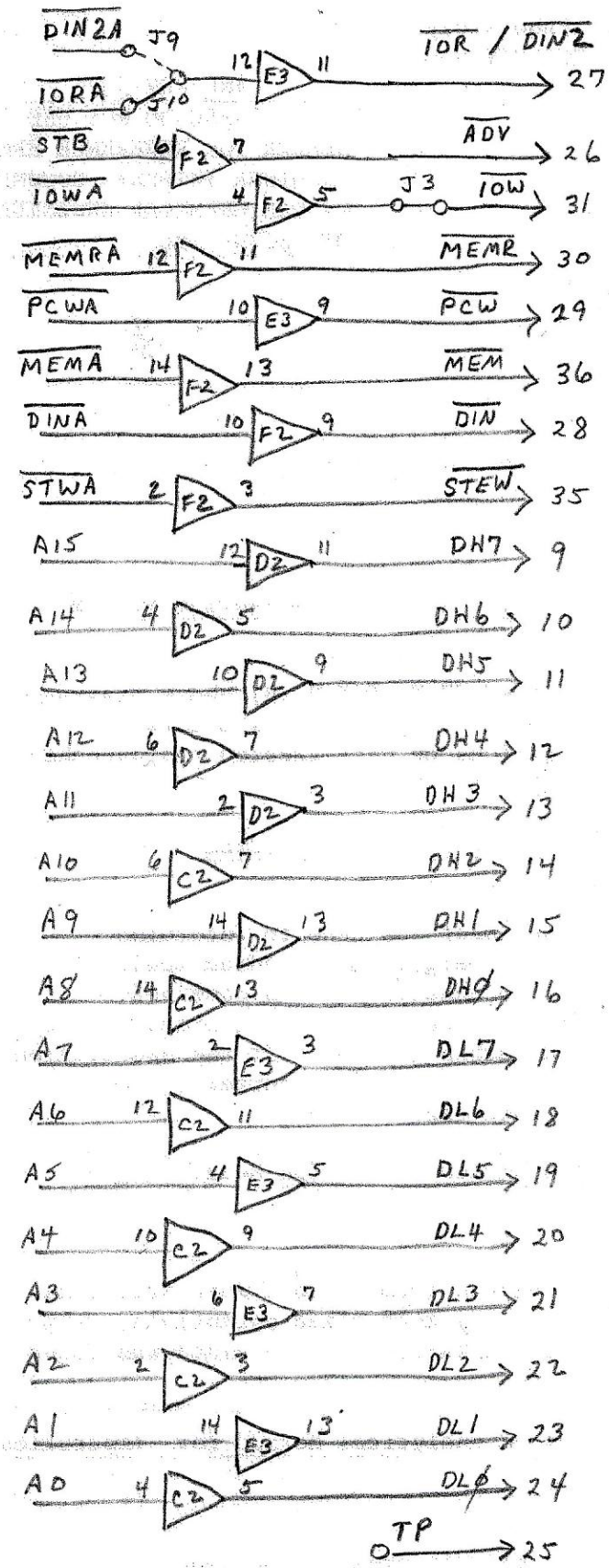




INTERRUPT LOGIC



ADRESS/CONTROL BUS DRIVERS



TYPICAL ENABLE OF
C2, D2, F2, E3

OPTIONAL FOR OVERTONE XTLS